COMPUCOLOR AND INTECOLOR USER'S BULLETIN

VOL. 4 NO. 3 $15.00 PER ISSUE

ISSUE NO. 15

COPYRIGHTED 1982 BY STAN PRO

PAGE	ARTICLE	AUTHOR
1	INDEX	F. Inally
2	EDITORS CORNER	STAN PRO
5	HOW TO RECAPTURE LOST DOCUMENTS	STAN PRO
6	EPSON GRAPHICS	MARTIN REX
9	EPSON MODIFICATIONS	STAN PRO
10	EPSON LETTER	JIM FARQUHAR
11	INTECOLOR 2/79 AND 5/79 INTERNAL FEATURES	A. NOMOMUS
13	JOYSTICKS (UPDATE)	S. PERRIGO
15	HOW TO USE RS232 TX READY JUMP TABLE	TOM GREEN
16	CONSTANTS, TABLES, VECTORS AND TEXT FOR CCII VER. 8.79 (BASIC DECODED)	TOM GREEN
20	GRAPHICS CLIPPING ROUTINE PART 2	TOM HUDSON
21	BASIC FILE SPACE REMOVAL UTILITY SOURCE CODE	F. ILES
26	NEW CLUB RELEASES	WORKING LATE
30	NEW KEYBOARD ROUTINE	R. CCHREINER

PROGRAM WINNERS ANNOUNED. LOTS OF WINNERS ARE LISTED
WE NEED ARTICLES FOR THE NEXT ISSUE. WE ALSO NEED MORE ARTICLES FOR THE NEXT ISSUE. ASIDE FROM THAT WE COULD USE SOME ARTICLES.

COMPUCOLOR & INTECOLOR
USER'S GROUP

S.P.
ELECTRONIC SYSTEMS

5250 Van Nuys Blvd.
Van Nuys, Calif. 91401
788-8850

VAN NUYS, CA 914

Robert Millikan
37c USA

RETURN POSTAGE
GUARENTEED

FIRST CLASS

[X] IF THIS BLOCK IS CHECKED YOUR SUBSCRIPTION EXPIRES WITH THIS ISSUE. US RENEWAL $35.00, FOREIGN $45.00
YOUR MEMBER NO. IS 320 PLEASE REFER TO IT IN ALL CORRESPONDENCE.

EDITOR'S CORNER

BY STAN PRO

EPSON PROBLEMS

Epson has finally gotten rid of the square wave handshake and that takes care of a major problem. There are still some CCII interface problems with the graphics. We are offering some of our solutions to these problems in this issue.

EPSON GRAPHICS

We have recieved an Epson Utilities disk from Martin Rex which does Graphics on the Epson. A copy of the graphics is enclosed in the bulletin. It's excellent as far as it goes but it does have limitations. What would be very useful would be a program either in Assembly or in Basic that would call from memory using 6 bytes per character the special character set in the CCII. In the Grafics Driver we wrote for the Base 2 we first looked at the Status bit to determine if we were looking at a graphic or ASCII character and then went to the proper place in the table with a call to a memory location equal to an offset + 6 x the ASCII value. The six required bits to generate the proper character were then called and sent to the printer buffer. I don't know if it would be quicker to use this method in the graphics mode or use the correspondence between the CCII graphic values and the required values to drive the EPSON hammers to generate the proper codeing with anding and oring mathmatically.

DISKS WE OWE

We are now all caught up on all the disks that we have been paid for that we are required by law to send out with the utmost speed. As soon as this bulletin goes out and I get a Resume' typed up I will send out disks to every one that has submitted a disk to the group. If you don't get a disk within 30 days after receiving your bulletin please drop us a reminder as your disk may not have reached us or reached us and dropped thru the cracks. Over the last 5 years I think I have developed a fairly good reputation for returning disks that have been submitted and I don't want to blow it now.

ADVERTISEMENTS

Any one who has some good programs they want to peddle we will exchange free space in this bulletin for free copies of the program. We remain the sole arbiters of how much space for which programs. That offer goes even for the programs we already have copies of thru member submission or used system purchase (When we buy someone's Computer they generally toss in the software they have). Seeing as we have already purchased most all ISC programs they must be exempted.

CONTEST

It was very hard to pick the best program. Doug Peel's Invaders was certainly one of the best but that was sold by us on a royalty basis and really wasn't a true submission. The Invaders submitted by the Australian group was very good but came after Doug's. Overall I would have to pick Asteroids, by Tom Hudson because, 1) It was very well documented with source listings, 2) It was of Arcade quality, 3) It was very very good, 4) I liked it best, 5) I still don't see how he got that much action into it with just one processor. Tom Hudson, you have won a set of paddles or a memory kit. Tom submitted a number of other excellent programs, a number of which where almost as good as Asteroids. Tom, if you will get in touch with me I will see that you get your prize.

RUNNER UP
We had not promised a runner up but Bill Rhodes certainly deservers a runner up prize for his additions to the print driver and his other contributions. Bill please get in touch.

ADDITIONAL PRIZES
There have been a number of very heavy contributers to the library. We will award them in addition to the programs we already sent them in return for their submissions either 3 diskfuls of programs from around the world (our choice) or the opportunity to purchase a 16k Memory Kit for something below our cost $ 50. These special contributors are Don Miller, Amod, Martin Rex, Ralpharythms, Rick Manazir, Roy Schaffer, Steve Perrigo, J. Neuman, and B. F. Muldowney. Its 4 am and I'm tired. If I spelled any names wrong (Steve) I'm sorry.

THE HOOK
Their is a hook in the unpromised prizes. You must still own your Compucolor and must be interested in receiving the prize. To receive your prize you must notify me in writing or by phone of your interest and give me your latest address to send the prize out to. For practical purposes a time limit of Dec 31, 1982 will be established to notify me of your latest address. Runner up and Additional prizes are what we consider the unpromised prizes.

MOST PROGRAMS
We have seen a number of ISC distributors advertising that they have the largest selection of programs available fof ISC machines. I have seen a number of their catalogues and they have many fine programs. Unfortunately, some of their disks are right out of our library. As to size, some catalogues are quite extensive but I haven't seen any that have 125 disks in them. Thats about how many Compucolor disks we've listed to date and we have another 75 or so to add to the collection as we review them. In addition, we have about 50 disks of Intecolor programs, some in three different software versions.

D. PEEL
First, I don't see how you do it for the price you charge. Second, I hate to say I told you, but I haven't sat down at my CCII just to enjoy it and play some games for a long time. The thing seems to grow and it eats up all the time you can spend.

S. P.
ELECTRONIC SYSTEMS
5250 VAN NUYS BLVD.—788-8850
VAN NUYS, CALIFORNIA 91401

By the time I get to start using it for my own pleasure, it probably won't work. One more thing - I will be glad to publish any good articles you may have.

ISC RELIABILITY

If you have had problems with the reliability of the equipment you have purchased from ISC, or with their service or repairs, please contact me.

LEGAL CENSORSHIP

We are sorry we had to withold a very interesting article in spite of the interest expressed in it. Hopefully we will be able to present it later.

STOCK REPORT

We hear ISC's sales are or were down 10%. No verification of this, though, as we have not seen the article ourselves.

FORTH

We have Forth up and running in our Compucolor. Not promised or soon to be delivered, but running. The author would like to know how much interest there is in it. He would like to put it in a Prom and sell it if there is enough interest. Please inform us if you are interested and how much you would be willing to pay for it.

SPEECH

Our CCII talks. Using a Vocoder and 8 other IC's we have secured a complete voice system for our CCII. The speech program is written in Forth andincludes a vocabulary of about 75 words. More are easiliy added. I was even able to add some. Forth programs can be written for it to make it count and do all sorts of interesting things. Steve Lewis, you've done one hell of a job, but the thing to do now is make it work in Basic. If Steve delivers, we will present a complete schematic in the next issue. No promises, though.

CCIII

Yes, there is a CCIII and we do have some information on it. I do not feel that it would be proper to publish it at this time. Perhaps in the next issue. (Yes, it does come from Italy.)

SPEAKER ON CCII

While it is true that you can get all sorts of speaker kits for the CCII, there is an easier way that seems to work. One of our former employees, Jim, hooked his up with a 470 ohm resistor in series with the primary of a small audio output transformer and hooked the speaker up to the secondary. No power, except as put out by the chip. It worked fine.

RELAY ON OUTPUT

We have seen a number of systems for connection of external relays to the output of the CCII. There are lines that go into the disk drives. Without the enable they don't do anything. We connected one of these lines up to a relay driver IC and with one simple chip we were able to easily, on command, open and close the relay. I am not absolutely sure but I think there is one unused line into the disk drive that can also be used. No promises, but if we can find the schematic we will put it in the next issue.

PAPER TIGER GRAPHICS
Been available for over a year on one of our disks.

MEMORY UPGRADE
The 16k memory kits are easy to add. We suggest that if your machine started life as an 8k, don't try to unsoldier the chips unless you're an expert.

LISTINGS
We are going to start presenting listings but they will be for programs of special interest. We have some for graphic printouts and some source codes that reveal many internal locations. The first one appears in this issue. There are more to come. It will be 2 am before th printer is finished with this so anything else will be saved for the next issue.

HOW TO SAVE HOURS OF WORK OR RECAPTURE A LOST DOCUMENT

BY STAN PRO

While working on Editor's corner at around 2: 30 am in the morn- I screwed up and blew the final draft. Went to same it and blew it out. There's no way I was going to start remembering all the things I had said in it. (As if they are worth remembering.) Well blowing it out on the disk at least leaves it in memory but how to get to it. It was late but I came up with the little program shown below.

The idea is to scan the entire memory (Not being sure where the buffer started I just started from scratch and searched thru memory. Every time the program encounters an ASCII character it sends it out to the printer.

There was one little problem. Some times the buffer had carraige return-line feeds and sometimes only one. I am not sure why and I was to tired to figure out why so I just sent out both a 10 and a 13 every time either was encountered. This might require some experimenting with to suit individual cases.

This gave a rough draft of every thing in the buffer and the morn my assistant was able to retype the column in a short time with nothing lost.

```
5 PLOT 27,77:OUT 8,4
10 FOR N= 36864TO 6555510FOR N= 36864TO 65535
20 Y= PEEK (N)
30 IF Y> 31AND Y< 128THEN GOSUB 100
35 IF Y= 13OR Y= 10THEN GOSUB 200
40 NEXT
50 POKE 33265,0
60 OUT 8,255
100 OUT 6.Y
110 RETURN
200 OUT 6,10
210 OUT 6,13
220 RETURN
READY
```

EPSON by Martin Rex

GRAPHICS DISPLAY MADE BY STAN WITH REX PROGRAM

I have recently purchased an Epson MX80 printer for my CCII. Here are a couple of tidbits that anyone thinking about the MX80 should know.

1. The friction feed adaptor kit sold by Orange Micro and others does not work. The paper tends to slip on every other linefeed or so. The kit provides only one set of rollers which is not sufficient to keep the paper from slipping. (Selectric typewriters have 2 rollers.) I suggest you consider the MX80FT if you need friction feed.

2. The new high resolution graphics PROM does not work with the 8141 serial board that is usually sold to those asking for the serial interface. If you ever plan to use the graphics capabilities of the printer, make sure you get the 8150 serial board with the 2K buffer. The 8150 costs $135-149 compared to $75 for the 8141.

3. So far I have been unsuccessful in writing a BASIC program for the 8141 to print the screen display if the display contains the CCII special graphic characters. (The Epson PROM contains the TRS80 graphic characters made up of 6 blocks while the CCII special contain 8 blocks.) Theoretically, it can be done easily by simply printing the first top 6 blocks of the CCII character set with the standard Epson set in one pass, scrolling the paper up the height of one block, then printing the bottom 2 blocks in a second pass. This can be done one full line at a time to save time. But, here is the hitch. The Epson is programmed to always form-feed the full 1/8" after a graphic character is printed even if the printer is set for 1/72" line-feeds. Therefore the printer prints a blank line after any line that contains a graphic character, merrily lousing up any display that has vertically contiguous graphic characters. The solution is the approach taken by the author of the article in the last FORUM where the screen is scanned three pixels at a time then printed (would you publish that program for us), or perhaps here's a better solution.

With the 8150 serial card and the high res graphics PROMS in the Epson, a program can be written that converts any CCII characters that the Epson doesn't understand into a high res character in a 6x8 dot matrix. This character would be stored in a data statement. I've encluded a program that does just that. It runs fairly fast at 4800 Baud. One full line of data is sent to the printer buffer at a time. While that line is being printed, the computer is busy decoding the next line. This particular program decodes only the CCII special block characters, but with a few more lines of code, anything that the CCII will print to the screen could be printed.

By the way, there is no problem at all in dumping the screen to the MX80 using the 8141 and the standard EPROMS if there are no graphic characters since all the other ASCII numbers are the same. You will find encluded a basic program called 'SCNDMP' which does it with just 2 lines of code.

Obviously, a machine program to do all this is the best solution of all. I understand that Eike Mueller of Intelligent Computer Systems 12117 Comanche Trail Huntsville, Alabama 35803 has written a machine language program to do a screen dump on the IDS Paper Tiger using high * resolution graphics. He tells me that as soon as he can get a graphics PROM for his MX80, he will write the program for the Epson. This method is much better since Eike is planning to include in the program the ability to increase or decrease the size of the display as it is printed.

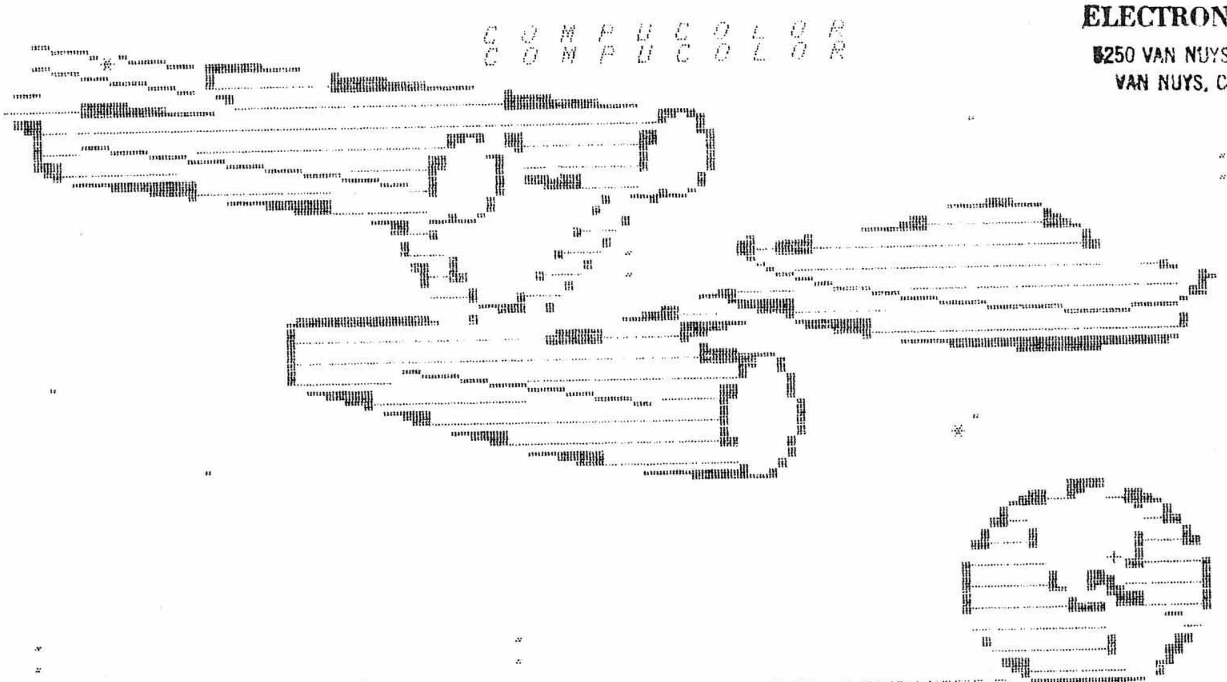
*AVAILABLE ON SP DISK FOR ABOUT A YEAR

4. I have included a text file that identifies the proper connections on the RS232 cable.

I would like you to send any programs that you have written for the Epson MX80. I am especially interested in either the word prossessor or the printer driver.

Take care,

Martin P. Rex



THIRD CENTURY AMERICA NEEDS HOME COMPUTERS

```
READY

LIST:

60000 REM  "GRFDMP.BAS;2"  GRAPHIC SCREEN DUMP WITHOUT REM
60010 REM   BY    MARTIN P. REX    8/9/81
60020 REM   SEE "GRFDMP.BAS;1"  FOR REMARKS
60030 DIM C(128),S(128),B1(255),B2(255)
60050 FOR N= 0TO 255:READ B1(N),B2(N):NEXT
60060 PLOT 14,27,18,6,27,13,27,51,24,13
60090 FOR L= 28672TO 32640STEP 128:FOR P= 0TO 127STEP 2
60110 IF PEEK (L+ P+ 1)< 128THEN C(P)= PEEK (L+ P):S(P)= 0:GOTO
60130
60120 C(P)= 32:S(P)= PEEK (L+ P):FLAG= 1
60130 NEXT P
60140 FOR N= 0TO 127STEP 2:PLOT C(N):NEXT
60170 IF FLAG= 0THEN 60240
60180 PLOT 13,27,75,128,1:FOR N= 0TO 127STEP 2
60220 PLOT B1(S(N)),B1(S(N)),B1(S(N)),B2(S(N)),B2(S(N)),B2(S(N))
:NEXT
60240 PLOT 13,10:FLAG= 0:NEXT L
60260 POKE 33265,0:PLOT 15:END
60290 DATA 0,0,192,0,48,0,240,0,12,0,204,0,60,0,252,0
60300 DATA 3,0,195,0,51,0,243,0,15,0,207,0,63,0,255,0
60310 DATA 0,192,192,192,48,192,240,192,12,192,204,192,60,192
60320 DATA 252,192,3,192,195,192,51,192,243,192,15,192,207,192
60330 DATA 63,192,255,192,0,48,192,48,48,48,240,48,12,48
60340 DATA 204,48,60,48,252,48,3,48,195,48,51,48,243,48
60350 DATA 15,48,207,48,63,48,255,48,0,240,192,240,48,240
60360 DATA 240,240,12,240,204,240,60,240,252,240,3,240,195,240
60370 DATA 51,240,243,240,15,240,207,240,63,240,255,240,0,12
60380 DATA 192,12,48,12,240,12,12,12,204,12,60,12,252,12
60390 DATA 3,12,195,12,51,12,243,12,15,12,207,12,63,12
60400 DATA 255,12,0,204,192,204,48,204,240,204,12,204,204,204
60410 DATA 60,204,252,204,3,204,195,204,51,204,243,204,15,204
60420 DATA 207,204,63,204,255,204,0,60,192,60,48,60,240,60
60430 DATA 12,60,204,60,60,60,252,60,3,60,195,60,51,60
60440 DATA 243,60,15,60,207,60,63,60,255,60,0,252,192,252
60450 DATA 48,252,240,252,12,252,204,252,60,252,252,252,3,252
60460 DATA 195,252,51,252,243,252,15,252,207,252,63,252,255,252
60470 DATA 0,3,192,3,48,3,240,3,12,3,204,3,60,3
60480 DATA 252,3,3,3,195,3,51,3,243,3,15,3,207,3
60490 DATA 63,3,255,3,0,195,192,195,48,195,240,195,12,195
60500 DATA 204,195,60,195,252,195,3,195,195,195,51,195,243,195
60510 DATA 15,195,207,195,63,195,255,195,0,51,192,51,48,51
60520 DATA 240,51,12,51,204,51,60,51,252,51,3,51,195,51
60530 DATA 51,51,243,51,15,51,207,51,63,51,255,51,0,243
60540 DATA 192,243,48,243,240,243,12,243,204,243,60,243,252,243
60550 DATA 3,243,195,243,51,243,243,243,15,243,207,243,63,243
60560 DATA 255,243,0,15,192,15,48,15,240,15,12,15,204,15
60570 DATA 60,15,252,15,3,15,195,15,51,15,243,15,15,15
60580 DATA 207,15,63,15,255,15,0,207,192,207,48,207,240,207
60590 DATA 12,207,204,207,60,207,252,207,3,207,195,207,51,207
60600 DATA 243,207,15,207,207,207,63,207,255,207,0,63,192,63
60610 DATA 48,63,240,63,12,63,204,63,60,63,252,63,3,63
60620 DATA 195,63,51,63,243,63,15,63,207,63,63,63,255,63
60630 DATA 0,255,192,255,48,255,240,255,12,255,204,255,60,255
60640 DATA 252,255,3,255,195,255,51,255,243,255,15,255,207,255
60650 DATA 63,255,4,4
```

EPSON MODIFICATIONS

BY STAN PRO

Some people myself included seem to be having trouble with the newer MX80FT's Epsons. We have had a lot of trouble with the feed when using the sprocket feed. The paper jams, twists and does about everything else imaginable. I have been less than satisfied.

The second problem I have encountered is to do graphics properly you need 8 bits. The problem is with the 2k buffer and Grafix 8 bits only works at 4800 baud. At 9600 all you get is garbage. On the other hand if you use 7 bits in your normal work (which is good enough for regular printing.) you can use the normal 9600 baud which doesn't requite special baud rate setting. A great convience. It also turns out if you want italics you also need the 8th bit. This seems to present a delema, as it is a pain to take the cover off every time you want to change settings.

The answere is an external switch. Fortunetly the Epson is hooked up so that most features are gotten by grounding various lines thru the internal switches.

By using a single pole, double throw swith that grounds either the 9600 baud line or the 8 bit line the unit may be externally thrown from 7 bit, 9600 baud to 8 bit, 4800 baud. While putting swithes on one was added to externally switch the printer from regular characters to italics.

When operating from a word processor or printing posters I prefer to have the program control page feeds. On the other hand when doing listings the CCII internal driver doesn't have paging. Ergo another switch to turn on the printers internal Skipover.

A number of the other functions can also be picked up by external swithches. These switchs greatly extends the capability and usefulness of the printer for me. [ALWAYS CHANGE SWITCHES WITH PRINTER OFF]

Jim this may be the answere to you problem. Why not give it a try.

By the way Jim lives in Australia and uses his Epson on 50 cycle current. When we sold it to him we wouldn't gaurentee it to work on 50 cycles so we were pretty glad to hear everything worked out O K.

One other comment. I don't understand what all the fuss is about. I keep reading about the little ' y ' s. As you will note there are none in the listing in this bulletin or in this printout. We suppress them with an OUT 8,4 and sent the the text to the printer with a PLOT 27,77. Control is regained with a POKE 33265, 0 and OUT 8, 255.

I guess we can also tell how we solved the problem of the flashing ready signal. The interface we developed has a memory cell to remember the last character sent. It also has a gate to block the pulses coming from the Epson. When the Epson starts sending hand shake signals to the CCII the first gets thru. The gate

S. P.
ELECTRONIC SYSTEMS
8250 VAN NUYS BLVD. 788-885
VAN NUYS, CALIF

then blocks any further pulses. The Epson can now sit there all day and send garbage to the CCII and it wont see it. However as soon as some legitimet data is sent the hand-shake gate opens up to allow the next real handshake signal thru. This may sound simple and it is but the developement that went into it and the technicians time in building and testing the device was quite extensive. The device we were selling replaced the Epson RS232 interface. The reason we developed it was that we had ordered a dozen Epson from the distributer and then found that they were being retailed all over town at below our cost. We lost a lot of money on the deal but not as much as if we had to eat all those Epson's and then have problems with machines we sold at below our cost. The only way we came out ahead on the deal was before we came out with the interface we spent one hell of a lot of time assisting our customer with the wild y problem. What it did to a Compucolor was nothing to what it did to an Inte-Color. They really went wild. After we came up with our interface other than one unit that came loose in shipping, (We believe.) Every unit sold worked with no problems, saving us hours on the phone. It's 4. 36 am. To bed.

S. P.
ELECTRONIC SYSTEMS
5250 VAN NUYS BLVD.—788-8850
VAN NUYS, CALIFORNI 91401

Stan Pro
S. P. Electronics
5250 Van Nuys Blvd
VAN NUYS CA. USA 91401

Dear Stan,

I received the printer some time ago and the disk and serial interface instructions about ten days after. Thank you for the trouble I put you to.

You will be pleased to know that the printer works like a dream. I cleared it through customs, bought a step-down transformer and some paper, plugged it in and away it goes. No problems at all. I did have trouble though with the disk you sent me. I suspect it may have been X-rayed on the way into Australia and it is unreadable. I cannot even get a directory on it. However, I won't worry about that because I have access to printer routines through various clubs here.

The only other trouble I have had is in trying to use the GRFDMP routine published in Forum Vol 1, No 4. For some reason it doesn't work. It starts printing the screen, then prints a few C's and 0's then starts form feeding the paper through the printer. It looks as though it should work, but I can't figure out why it doesn't. You may like to mention this problem in your bulletin to see if anyone else has experienced this and has a cure.

Thanks again Stan. Keep up the good work.

Yours faithfully,

(Jim Farquhar)

A.1 Key Memory Locations

DECIMAL	HEX	DESCRIPTION
32768 TO 40447	8000-9DFF	Screen refresh RAM (48 line)
41353	A189	Points to maximum RAM used by BASIC
41393	A1B1	Points to start of BASIC source
41395	A1B3	Points to end of source and start of variables
41397	A1B5	Points to end of variables and start of arrays
41399	A1B7	Points to end of arrays
40851	9F93	User ESCAPE jump vector
40857	9F99	User FLAG jump vector 31
40854	9F96	User Flag jump vector 30
40848	9F90	User time no 2 jump vector
40953	9FF9	Keyboard FLAG
40884	9FB4	FCS output FLAG
40949	9FF5	Input port FLAG
40886	9FB6	BASIC output FLAG
None Exists	None Exists	Output port FLAG
40884	9FB4	LIST output FLAG
40955	9FFB	Keyboard character
40956	9FFC	Keyboard character ready FLAG
40832 TO 40847	9F80-9F8F	Printer Driver RAM
40832	9F80	Selected Printer
40833	9F81	Lines per Page
40834	9F82	Vertical Margin
40835	9F83	Horizontal Left Margin
40836	9F84	Physical Device Width
40837	9F85	Nulls/Spinwriter Pitch
40838	9F86	Current Line Number

DECIMAL	HEX	DESCRIPTION
40839	9F87	Current Column Number
40840	9F88	Spaces Used (Spinwriter)
40841	9F89	Lines Used (Spinwriter)
41421	A1CD	Location of CALL(x) jump
41424	A1D0	BASIC output vector location
41574	A266	Start of BASIC source code
65535	FFFF	Maximum RAM address

A2 Port Assignments

DECIMAL	I/O PORT ADDRESS	HEX
0 - 15	1st TMS 5501	0 - F
16 -31	2nd TMS 5501	10 - 1F
248	Cursor X Register (write only)	F8
249	Cursor Y Register (write only)	F9
250	Roll Register (write only)	FA
251	Light Pen Status	FB
252	Light Pen X Coordinate	FC
253	Light Pen Y Coordinate	FD
254	Light Pen Acknowledge	FE
128	Mini Disk Status	80
129	Mini Disk Readtrack	81
130	Mini Disk Readsector	82
131	Mini Disk Read Data	83
132	Mini Disk Write Command	84
133	Mini Disk Write Track	85
134	Mini Disk Write Sector	86
135	Mini Disk Write Data	87
136	Mini Disk Write Drive	88
137	Mini Disk Read Select Interrupt	89
138	Mini Disk CPU Hold	8A
160	Floppy Disk Status	A0
161	Floppy Disk Readtrack	A1
162	Floppy Disk Readsector	A2
163	Floppy Disk Read Data	A3
164	Floppy Disk Read Command	A4
165	Floppy Disk Write Track	A5
166	Floppy Disk Write Sector	A6
167	Floppy Disk Write Data	A7
168	Floppy Disk Write Drive Select	A8
169	Floppy Disk Read Interrupt	A9
170	Floppy Disk CPU Hold	AA

S. P.
ELECTRONIC SYSTEMS
8250 VAN NUYS BLVD.—788-885[illegible]
VAN NUYS, CALIFORN 91401

DECIMAL		TMS 5501 I/O CHIP	HEX	
1st 5501	2nd 5501		1st 5501	2nd 5501
0	16	Read Serial Data in from RS-232C Interface	0	10
1	17	Read Parallel Data from keyboard or 2nd 5501	1	11
2	18	Read Interrupt Address on TMS 5501	2	12
3	19	Read Status on TMS 5501	3	13
4	20	Issue Discrete Command	4	14
5	21	Set Baud Rate on Serial I/O	5	15
6	22	Transmit Serial Data to RS-232C Interface	6	16
7	23	Transmit Parallel Data to keyboard or 2nd 5501	7	17
8	24	Load Interrupt Mask Register	8	18
9	25	Interval Timer #1	9	19
10	26	Interval Timer #2	A	1A
11	27	Interval Timer #3	B	1B
12	28	Interval Timer #4	C	1C
13	29	Interval Timer #5	D	1D
14	30	No Function	E	1E
15	31	No Function	F	1F

JOYSTICKS by Steve Perrigo

After writing and running some double joystick routines I discovered that my selection of joystick polling positions were't the best. Apparently, double angle closures with a common vector can cause erroneous readings (a sort of echo from one joystick to the other). I've worked up an alternate system that appears just as useable. However, before anything is published about it in FORUM I would appreciate a critical look at the logic behind it by you or another 'interested party' up there. My only reservations about this new proposal is that it changes the fire button on the primary joystick from 5 to 0 and that the positions for the secondary joystick are not really useable on the keyboard. Eike is sending me a review of the polling positions used by his single joystick so at the moment I don't know which key his fire button addresses. Anyway - take a look and let me know what you think. Here is a table and a sample basic listing (modification of the one published) to read the keyboard. The listing has an added feature of being able to return both a direction and detect a fire button closure 'simultaneously' which would be good for fast action games.

JOYSTICK CHARACTER INFORMATION TABLE

S. P.
ELECTRONIC SYSTEMS
[illegible]250 VAN NUYS BLVD.—788-885[illegible]
VAN NUYS, CALIFORNI 91401

DIRECTION	CHR$	ASCII	OUT(7)		INP(1)	
FOR JOYSTICK 0						
FIRE	0	48	15	0F	254	FE
DOWN	2	50	13	0D	254	FE
LEFT	4	52	11	0B	254	FE
RIGHT	6	54	9	09	254	FE
UP	8	56	7	07	254	FE
FOR JOYSTICK 1						
FIRE	A	65	14	AE	253	FD
DOWN	C	67	12	0C	253	FD
LEFT	E	69	10	0A	253	FD
RIGHT	G	71	8	08	253	FD
UP	I	73	6	06	253	FD

```
100 JS=0: GOSUB 1000: REM READ JOYSTICK 0
200 JS=1: GOSUB 1000: REM READ JOYSTICK 1
500 REM
510 REM LINES 100 AND 200 CALL THE SUBROUTINE TO READ
520 REM EACH JOYSICK.  A VALUE OF 0 TO 9 WILL BE RETURNED
530 REM IN 'T' AND A VALUE OF 0 OR 5 WILL BE RETURNED
540 REM IN 'F' INDICATING THE FIRE BUTTON AS NOT DEPRESSED

550 REM OR DEPRESSED, RESPECTIVELY.
560 REM
570 REM
1000 OUT 8,247:FOR TM =1 TO 100: NEXT
1010 T=0: F=0
1020 OUT 7,13-JS: IF INP(1)=255-JS THEN T=T-3
1030 OUT 7,11-JS: IF INP(1)=255-JS THEN T=T+5
1040 OUT 7,9-JS: IF INP(1)=255-JS THEN T=T+7
1050 OUT 7,7-JS: IF INP(1)=255-JS THEN T=T+9
1060 IF T>12 THEN T=T-6
1070 T=ABS(T)-1: IF T=-1 THEN T=0
1080 OUT 7,15-JS: IF INP(1)=255-JS THEN F=5
1090 OUT 8,255: RETURN
```

Stan – This is my own "rebuttal" to my article in FORUM about the dual joystick controller hook-up + routines. This system seems better – what do you think?

Steve

Note – this is from a letter
to Doug Peel

==== How to Use the RS-232 Tx Ready Jump Table ====

BY TOM GREEN

The Compucolor II has a limited facility for multiprocessing! The facility has a very low profile, but it IS there and it DOES work [it makes <ESC> <X> work]. Unfortunately, the facility is difficult to understand and use, which may explain ISC's silence on the subject. It takes advantage of the TMS-5501 chip's Interupt-Mode for output, combined with the RS-232 Tx Ready Jump Table, plus two pseudo-registers in the System Scratch Pad Memory, OUT.FL [the Out Flag, one byte at 81C3h] and OUT.HL [the Output Cursor Address, one word at 81C3h].

The main difficulty in understanding this facility is that of understanding the concept of interupts: The TMS-5501 chip has a "list" of 8 hardware events which it is supposed to look for. Whenever it notices one of these events, it tells the CPU to stop what it is doing, and execute a one-byte instruction of of the 5501's choice instead.

This instruction is almost always a RST, the 1-byte version of the CALL, so that eventually the CPU can RETurn to whatever it was doing when it was interrupted. There are eight RST intstructiones, each of which CALLs its subroutine at a specific location, called its Restart Location: RST 0's Restart Location is at 0000h, RST 1's is at 0008h ... and RST 7's is at 0038h. Few subroutines will really fit into the eight bytes between each location, instead these locations usually hold JMPs to the true subroutine locations.

Event #5 on the list of hardware events is "RS-232 Transmit Buffer Empty... TMS-5501 ready for a character to transmit". The TMS-5501 chip causes an RST 5 interupt whenever it notices this condition, and away goes the CPU to 0028h. There is a catch however: the operating system usually leaves the 20h-bit OFF whenever it sends a new "interupt mask" to the TMS-5501 with an "OUT 08h". If you've guessed that 20h-bit tells the TMS-5501 whether or not to notice event #5, you're right. If you were to send a new mask to the 5501 with the 20h-bit ON, the TMS-5501 would immediately cause a RST 5 interupt and transfer control to the TX.RDY Subroutine. TX.RDY would check OUT.FL to find out what to do. Normally OUT.FL holds 00h, telling TX.RDY to "re-mask RST 5 and return."

Not much help so far, granted ... BUT if you first place a 01h in OUT.FL, then when you un-mask that 20h-bit, the 01h will tell TX.RDY to transmit KBCHA [the Last Keyboard Character, 1 byte at 81FEh] before re-masking RST 5 and returning. Now we're making progress ... Place 06h in OUT.FL and TX.RDY will jump via J.OUSR [a JMP instruction at 81C2h] each and every time its ready transmit another character, interupting BUT NOT ENDING whatever other program you may running at the same time until OUT.FL is changed from 06h. Hmmm, does anyone have a print-spooler?

Output flag 02h seems to have been where the <ESC> <C> facility used to be. It now leads to a JMP 0000h, which in turn leads to the main Power-Up Routine.

The other Output Flags are used by XMT.PG [the <ESC> <X> processor] to control the <LF><CR> it transmits at the end of each line and the B0h it sends when it is finished.

=== Contstants, Tables, Vectors, and Texts for Ver 8.79 ===

BY TOM GREEN

```
J.RST0 0000h  JMP 01BAh: PWR.UP = Power-Up Reset Handler
@$RSET 0003h      1F38h: $RESET = [RESET] Entry to BASIC
       0005h        FFh: Filler
MODE.0 0006h        00h: Local  = Initial Terminal Mode
ROLL.0 0007h        01h: Scroll = Initial Scrolling Mode
J.RST1 0008h  JMP 01C8h: TM.USR = User-Timer Routine
@T.ESC 000Bh      008Dh: T.ESC  = <ESC> Sequence Jump Table
END.PG 000Dh        B0h:        = Marks "End of Transmission" from <ESC> <K>
       000Eh        0Dh:
       000Fh        FFh:
J.RST2 0010h  JMP 0253h: TICTOC = Real-Time-Clock Advance Routine
       0013h        FFh:
TAUCLK 0014h      0F00h:  8/15s = Interval between Real-Time-Clock Advances
BAUD.0 0016h        C0h: 9600bd = Initial Baud Rate
OK.INT 0017h        08h:        = TMS-5501 Cmnd: "Enable Interupt Acknowlege"
J.RST3 0018h  JMP 0504h: SVC.KB = Keyboard Polling Routine
TAUBEL 001Bh        F4h: -150ms = Starting point of Stop-the-Bell Countdown
RT.RPT 001Ch        05h:        = Repeat Key Rate
RT.KEY 001Dh        FFh: 16.3ms = Keyboard Polling Rate
@READY 001Eh      81FFh: READY  = Keyboard Character Ready Flag
J.RST4 0020h  JMP 03BDh: RX.AVL = RS-232 Input Routine
       0023h        40h:
J.KTST 0024h  JMP 0B18h: KY.TST = Fetch next keystroke character, if any.
RT.STP 0027h        14h:  6.9ms = Interval Required for Disk to Step 1 Track
J.RST5 0028h  JMP 04B8h: TX.RDY = RS-232 Output Routine
       002Bh  JMP 0398h:        = Unidentified Return Vector from SVC.KB
OK.C   002Eh        04h:        = TMS-5501 Intrpt Mask: "OK: Clock"
       002Fh        A0h:
J.BOUT 0033h  JMP 03B4h: BASOUT = Interface for Output from BASIC
STAK.0 0036h      8042h:        = Initial SP Value
J.RST7 0038h  JMP 01B6h: HO.HUM = Main Thumb-Twiddle Routine
       003Bh  JMP 050Dh:        = Unidentified Return Vector from SVC.KB
DEV0   0040h  03h Bytes: "CD0"  = Initial Default Device
J.HAN0 0043h  JMP 1AC1h: D.HAN0 = Disk Type 0 Handler Routine
D.TYP0 0046h  02h Bytes: "CD"   = Disk Type 0 Designation
D.NBR0 0048h        02h: 0:, 1: = Disk Type 0 Number of Disks Installed
D.SCT0 0049h        0Ah: 10 S/T = Disk Type 0 Number of Sectors/Track
       004Ah        50h:          Referenced at 1BD9h, possibly a timing fctr
DPB1   004Bh  08h x FFh: Filler = Disk Type 1 Parameter Block
DPB2   0053h  08h x FFh: Filler = Disk Type 2 Parameter Block
       005Bh      FFFFh:
T.CTL  005Dh  18h Addrs:        = Control Character Jump Table
                  0453h: IGN.KY = <^@> Handler  <NULL>
                  1F3Eh: $AUTO  = <^A> Handler  <AUTO>
                  07F7h: ST.PLT = <^B> Handler  <PLOT>
                  0452h: ST.FLG = <^C> Handler  <CUR-XY>
                  0453h: IGN.KY = <^D> Handler
                  0453h: IGN.KY = <^E> Handler
                  0452h: ST.FLG = <^F> Handler  <SET CCI>
                  051Ah: RG.BEL = <^G> Handler  <BELL>
                  02B7h: VC.HOM = <^H> Handler  <HOME>
                  02FAh: VC.TAB = <^I> Handler  <TAB>
```

```
T.CTL  cont.       0303h: VC.DN  = <^J> Handler  <Down Arrow>
                   0543h: ERS.LN = <^K> Handler  <ERS LN>
                   027Fh: ERS.PG = <^L> Handler  <ERS PG>
                   02BDh: VC.CR  = <^M> Handler  <RETURN>
                   0331h: VC.A7  = <^N> Handler  <A7 ON>
                   038Fh: VC.A7F = <^O> Handler  <BLK, A7 OFF>    <^P>....<^W>
                   04EDh: CRT.TX = <^X> Handler  <XMIT PG>        not in table
                   02FEh: VC.RT  = <^Y> Handler  <Rt Arrow>
                   0339h: VC.LT  = <^Z> Handler  <Lt Arrow>
                   0452n: ST.ESC = <^[> Handler  <ESC>
                   033Fh: VC.UP  = <^\> Handler  <Up Arrow>
                   0384h: FL.ON  = <^]> Handler  <FG / FLG OFF>
                   0383h: FL.OFF = <^^> Handler  <BG / FLG ON>
                   0388h: VC.BNK = <^_> Handler  <BLINK>
T.ESC 008Bh 20h x Addrs:         = <ESC> Sequence Jump Table
                   0453h: IGN.KY =  (@) Handler
                   0452h: ST.FLG =  (A) Handler  (BLND)
                   0446h: ST.PLT =  (B) Handler  (PLOT)
                   0453h: IGN.KY =  (C) Handler  (XMIT CUR)   In Specs, not Pgm
                   16FFh: FCS    =  (D) Handler  (FCS)
                   1F2Ch: $RSTRT =  (E) Handler  (BSC RST)
                   04A5h: ST.DPX =  (F) Handler  (DUPLEX)
                   16F4h: FCS232 =  (G) Handler  (FCS RS-232)
                   04A3h: ST.HLF =  (H) Handler  (1/2 DPLX)
                   9000h: UsrRAM =  (I) Handler
                   04E7h: ST.VRT =  (J) Handler  (VERTICAL)
                   04DCh: ST.SCL =  (K) Handler  (ROLL)
                   04A6h: ST.LCL =  (L) Handler  (LOCAL)
                   0447h: ST.232 =  (M) Handler  (TERM MODE)
                   0447h: ST.OFL =  (N) Handler  (SLEEP)
                   0453h: IGN.KY =  (O) Handler
                   4000h: No Mem =  (P) Handler  (CRT OP SYS)
                   0453h: IGN.KY =  (Q) Handler
                   0452h: ST.FLG =  (R) Handler  (BAUD RATE)
                   A000h: UsrRAM =  (S) Handler  ("ASMB")
                   8200h: UsrRAM =  (T) Handler  ("TEXT")
                   0453h: IGN.KY =  (U) Handler
                   0453h: IGN.KY =  (V) Handler
                   1F26h: $BASIC =  (W) Handler  (BASIC)
                   04DDh: ST.PAG =  (X) Handler
                   0452h: ST.FLG =  (Y) Handler  (TEST)
                   4800h: No Mem =  (Z) Handler  (45 DN)
                   0453h: IGN.KY =  ([) Handler
                   5000h: No Mem =  (\) Handler  (45 UP)
                   5800h: No Mem =  (]) Handler
                   81BFh: J.UINP =  (^) Handler  (USER)
                   01A7h: J.CRT  =  (_) Handler  (CRT)
T.OFLG 00CDh   08h Addrs:        = Output Flag Jump Table              See:
                   04B0h: TX.OFF = Flag 00h: Turn off Xmtr      "How to Use the
                   04AAh: TX.KB  = Flag 01h: Echo KB.CHR         Tx Rdy Jmp Tbl"
                   0561h: J.0000 = Flag 02h: [ Disabled ]
                   0569h: TX.COL = Flag 03h: Xmit ((OUT.HL)) or <LF> if Col 00h
                   059Ch: TX.CR  = Flag 04h: Xmit <CR>
                   05A5h: TX.C00 = Flag 05h: Xmit ((OUT.HL)) even if in Col 00h
                   81C2h: J.UOUT = Flag 06h: User Output Routine
                   05ADh: TX.EOP = Flag 07h: Xmit "End/Page"
T.IFLG 00DDh   20h Addrs:        = Input Flag Jump Table              Next Flag
                   0425h: OUT.VC = Flag 00h: Output via Visable Cursor  (00h)
                   0405h: OUT.BC = Flag 01h: Output via Blind Cursor    (01h)
                   0709h: OUT.PL = Flag 02h: Output via Plot Routine    (02h)
                   0454h: ST.CU2 = Flag 03h: Select SETVCX(05h) / SETBCA(07h)
                   0861h: OUT.PP = Flag 04h: Output via Plot w/ Pad Rtn (04h)
```

```
T.IFLG cont.       0468h: ST.VCY = Flag 05h: Set Visable Cursor Y Addr  (00h)
                   044Ah: ST.VCC = Flag 06h: Set Visable Cursor CCI     (00h)
                   0473h: ST.BCX = Flag 07h: Set Blind Cursor X Addr    (08h)
                   0480h: ST.BCY = Flag 08h: Set Blind Cursor Y Addr    (09h)
                   044Eh: ST.BCC = Flag 09h: Set Blind Cursor CCI       (00h)
                   81C5h: J.UINP = Flag 0Ah: Xfer Char to User Inp Rtn  (0Ah)
                   81C5h: J.UINP = Flag 0Bh: Xfer Char to User Inp Rtn  (0Bh)
                   0453h: IGN.KY = Flag 0Ch: Ingore Character           (0Ch)
                   1737h: INP.FC = Flag 0Dh: Xfer to FCS Input Routine  (0Dh)
                   17F9h: OUT.TX = Flag 0Eh: Output via RS-232 Tx Line  (0Eh)
                   0453h: IGN.KY = Flag 0Fh: Ignore Character           (0Fh)
                   81C5h: J.UINP = Flag 10h: Xfer Char to User Inp Rtn  (10h)
                   81C5h: J.UINP = Flag 11h: Xfer Char to User Inp Rtn  (11h)
                   04C6h: ST.BDR = Flag 12h: Set Baud Rate              (00h)
                   81C5h: J.UINP = Flag 13h: Xfer Char to User Inp Rtn  (13h)
                   81C5h: J.UINP = Flag 14h: Xfer Char to User Inp Rtn  (14h)
                   81C5h: J.UINP = Flag 15h: Xfer Char to User Inp Rtn  (15h)
                   81C5h: J.UINP = Flag 16h: Xfer Char to User Inp Rtn  (16h)
                   1F3Bh: $INPUT = Flag 17h: Xfer Char to BASIC Inp Rtn (17h)
                   81C5h: J.UINP = Flag 18h: Xfer Char to User Inp Rtn  (18h)
                   027Ah: OUT.SF = Flag 19h: Xfer Char to Screen-Fill R (00h)
                   81C5h: J.UINP = Flag 1Ah: Xfer Char to User Inp Rtn  (1Ah)
                   0501h: J.ESCS = Flag 1Bh: Xfer Char to <ESC> Seq Rtn
                   81C5h: J.UINP = Flag 1Ch: Xfer Char to User Inp Rtn  (1Ch)
                   81C5h: J.UINP = Flag 1Dh: Xfer Char to User Inp Rtn  (1Dh)
                   81C5h: J.UINP = Flag 1Eh: Xfer Char to User Inp Rtn  (1Eh)
                   81C5h: J.UINP = Flag 1Fh: Xfer Char to User Inp Rtn  (1Fh)
        011Dh  03h x FFh: Filler
        0120h  JMP 0AADh:        = Purpose Unknown: "MVI B,03h  RET"
CRTMSG  0123h  17h bytes: Text   = CRT Mode Greeting, w/ EFh Delimiter
        013Ah  10h x FFh: Filler = Could be used to extend CRT Greeting
$CPYRT  014Ah  5Dh bytes: Text   = BASIC Greeting & (c) Notice w/ 00h Delimiter
```

=== Executable Routines for Ver 8.79 ===

```
ST.CRT: 01A7h Entry: Action: Transfers control to the CRT Mode Software
        =NO=RETURN=


HO.HUM: 01B6h Entry: Action: Infinite Loop  =  Thumb-Twiddler for Main Exec
        =NO=RETURN=


PWR.UP: 01ABh Entry: Action: 1) Compares PUP [81B7h] to the constant 97h...
        =NO=RETURN=          2) If EQUAL:   performs a Warm Start and exits
                                            to ST.CRT  [81A7h]
                             3) If UNEQUAL: performs a Cold Start, setting
                                            PUP to 97h, and exits to $RESET
                                            [at 1F38h, pointed to by 0003h]
                             Note: 97h = 1001 0111b, which is a bit pattern
                                   which RAM memories do not usually assume
                                   when just-powered up.
```

```
GET.VC:                   Action: Fetches Visable Cursor Data from Scratch Pad Mem.

GET.V1: 0245h Entry: Result: SW = [ unchanged ]
                             A  = [ unchanged ]
                             B  = Visable Cursor X Address      [ 00h ... 3Fh ]
                             C  = Visable Cursor Y Address      [ 00h ... 1Fh ]
                             D  = Visable Cursor A7 Byte
                             E  = Visable Cursor CCI Status
                             HL = Screen RAM Addr corresponding to Vis Cur X&Y

GET.V2: 024Ah Entry: Result: SW = [ unchanged ]
                             A  = [ unchanged ]
                             B  = Visable Cursor X Address      [ 00h ... 3Fh ]
                             C  = Visable Cursor Y Address      [ 00h ... 1Fh ]
                             DE = [ unchanged ]
                             HL = Screen RAM Addr corresponding to Vis Cur X&Y
```

```
TICTOC: 0253h Entry: Action: Increments the Real-Time-Clock Registers


OUT.SF: 027Ah Entry: Action: Fills Screen with Test Char:  (ESC), (Y) Sequence
        =NO=RETURN=


ERS.PG: 02F7h Entry: Action: Erases Screen
        =NO=RETURN=
```

```
BC.HOM: 02ADh Entry: Action: Homes BOTH Cursors and un-scrolls screen.
        =NO=RETURN=


VC.HOM: 02B7h Entry: Action: Homes Visable Cursor.
        =NO=RETURN=


VC.CR:  02BBh Entry: Action: Returns Visable Cursor to Left Margin.
        =NO=RETURN=


XQTSCN: 02C2h Entry: Action: Xmits Cursor/Scroll Commands to SMC-5027 CRT Chip
                      Parms: C  = New X-Addr for Visable Cursor  [00h...40h]
                             B  = New Y-Addr for Visable Cursor  [00h...1Fh]
                       (81CDh)= New R-Addr for CRT Top Line      [00h...1Fh]
                     Output: SW = [ lost ]
                             A  = [ lost ]
                             BC = [ unchanged ]
                             DE = [ unchanged ]
                             HL = [ lost ]
                       (81CBh)= New X-Addr for Visable Cursor
                       (81CCh)= New Y-Addr for Visable Cursor
                       (81CDh)= [ unchanged ]
```

THIS IS THE SECTION OF TOM HUDSON'S ARTICLE OMITTED IN THE LAST ISSUE

A GRAPHICS CLIPPING ROUTINE BY TOM HUDSON

```
10 REM  SF IS INITIAL SIZE FACTOR
20 SF= 2
50 REM  SI IS STEP INCREMENT - AMOUNT TO INCREASE SIZE EACH DRAW cycle
LE
60 SI= 1.1
70 REM  RF IS ROTATION FACTOR - RADIANS TO ROTATE EACH PHASE
80 RF= .1
90 REM  CX AND CY ARE CENTER COORDINATES OF OBJECT ON SCREEN
95 CX= 64:CY= 30
110 PLOT 12
130 READ N:REM  N = # POINTS IN OBJECT
150 DIM X(N),Y(N),X2(N),Y2(N)
170 FOR X= 1TO N:READ X(X),Y(X):X(X)= X(X)* SF:Y(X)= Y(X)* SF:NEXT X
190 FOR RW= 0TO 6.28STEP RF
210 FOR X= 1TO N:X(X)= X(X)* SI:Y(X)= Y(X)* SI:NEXT X
230 FOR X= 1TO N:XW= X(X)* COS (RW)+ Y(X)* SIN (RW):YW= - X(X)* SIN (RW)+ Y
(X)* COS (RW)
250 X2(X)= XW* .75+ CX:Y2(X)= YW+ CY
270 NEXT X
300 FOR X= 1TO N- 1:X1= X2(X):Y1= Y2(X):X2= X2(X+ 1):Y2= Y2(X+ 1):GOSUB 100
0:NEXT X
310 X1= X2(N):Y1= Y2(N):X2= X2(1):Y2= Y2(1):GOSUB 1000
320 NEXT RW
400 GOTO 190:REM  LOOP FOREVER
500 DATA 4,10,10,10,-10,-10,-10,-10,10
999 REM  THE 1000 & 2000 ROUTINES CLIP THE SHAPE
1000 L1= 0:L2= 0:R1= 0:R2= 0:T1= 0:T2= 0:B1= 0:B2= 0
1005 IF X1< 0THEN L1= 1:GOTO 1020
1010 IF X1> 127THEN R1= 1
1020 IF Y1< 0THEN B1= 1:GOTO 1040
1030 IF Y1> 127THEN T1= 1
1040 IF X2< 0THEN L2= 1:GOTO 1060
1050 IF X2> 127THEN R2= 1
1060 IF Y2< 0THEN B2= 1:GOTO 1080
1070 IF Y2> 127THEN T2= 1
1080 IF L1+ L2= 2OR R1+ R2= 2OR T1+ T2= 2OR B1+ B2= 2THEN RETURN
1090 X3= X1:Y3= Y1:X4= X2:Y4= Y2:GOSUB 2000
1095 L1= L2:R1= R2:T1= T2:B1= B2
1100 X1= XW:Y1= YW:X3= X2:Y3= Y2:X4= X1:Y4= Y1:GOSUB 2000
1110 IF X1< 0OR X1> 127OR Y1< 0OR Y1> 127OR XW< 0OR XW> 127OR YW< 0OR YW> 1
27THEN RETURN
1120 PLOT 2,X1,Y1,242,XW,YW,255:RETURN
2000 IF L1+ T1+ R1+ B1= 0THEN XW= X3:YW= Y3:RETURN :REM  OK, ON SCREEN
2100 IF L1THEN XW= 0:YW= Y3+ (Y4- Y3)* (- X3)/ (X4- X3):X3= XW:Y3= YW:IF Y3
= > 0AND Y3< = 127THEN RETURN
2110 IF R1THEN XW= 127:YW= Y3+ (Y4- Y3)* (127- X3)/ (X4- X3):X3= XW:Y3= YW:
IF Y3= > 0AND Y3< = 127THEN RETURN
2120 IF B1THEN YW= 0:XW= X3+ (X4- X3)* (- Y3)/ (Y4- Y3):X3= XW:Y3= YW:IF X3
= > 0AND X3< = 127THEN RETURN
2130 IF T1THEN YW= 127:XW= X3+ (X4- X3)* (127- Y3)/ (Y4- Y3):X3= XW:Y3= YW:
IF X3= > 0AND X3< = 127THEN RETURN
2200 RETURN
READY
```

```
FCS>PRINT COMPAC

;
;       BASIC FILE SPACE REMOVAL UTILITY
;                   V1.00
;
; *** INTECOLOR 8001 V2.79 & V5.79 SPACE REMOVER V1.00 ***
; --------------------------------------------------------
; ******* COMPUCOLOR II V6.78 SPACE REMOVER V1.00 ********
;
;
; ********* MODIFIED  7/31/79  JKP III **********
;
; *******************************************************
; ********** MODIFICATIONS AND REVISIONS ****************
; *                                                     *
; * 5/11/79     MODIFIED FOR COMPUCOLOR II - V6.78      *
; *              'CI' INCLUDED;  JKP III                *
; * 5/14/79     INITIALIZE 'DROPS' COUNTER              *
; * 6/8/79      MODIFIED FOR CONDITIONAL ASSEMBLY FOR   *
; *              BOTH COMPUCOLOR II V6.78 AND           *
; *              INTECOLOR V2.79 AND V5.79              *
; * 6/11/79     SIMPLIFY 'CI' ROUTINE  JKP III          *
; * 6/25/79     IMPROVE 'BACKSPACE' ROUTINE, INCLUDE    *
; *             'BREAK' ROUTINE (FCS), RE-INSTITUTE     *
; *             'CHRINT'  JKP III                       *
; * 6/28/79     ADD CAPABILITY FOR DIFFERENT SOURCE AND *
; *              DESTINATION FILES   JKP III            *
; * 7/10/79     ADD 'EXPERIMENTAL' OPTION   JKP III     *
; * 7/31/79     RELEASED TO INTECOLOR   JKP III         *
; *              SR-8000-221  972008  COMPAC.PRG  V1.00 *
                 [illegible]ED TO COMPUCOLOR   JKP III  *
                 [illegible]21  972001  COMPAC.PRG  V1.00 *
                                                        *
           [illegible]  *************************************



        [illegible]ONFIGURATION'


       [illegible]  '  1 - INTECOLOR 8001  V2.79 & V5.79'
       [illegible]     2 - COMPUCOLOR II  V6.78'
       .PRINT ''
       .PRINT 'CRT TYPE '
       .INPUT CRTYPE
       .PRINT ''
       .PRINT ''
       .PRINT 'EXPERIMENTAL VERSION (0=NO, 1=YES) '
       .INPUT EXPMNT
       .PRINT ''
```

```
         .PRINT ''
         ENDIF
;
;
         IF CRTYPE EQ 2
;
;        ** SYSTEM ADDRESSES **
;
PRORG    EQU     8200H   ;** COMPUCOLOR II  V6.78 **

LO       EQU     3392H   ;SEND CHAR TO SCREEN
OSTR     EQU     33F4H   ;SEND STRING ENDING WITH 239

EMESS    EQU     262DH   ;FCS ERROR SUBROUTINE
PFSPC    EQU     3077H   ;PARSE FILE SPEC
RESET    EQU     26A5H
OPEN     EQU     2DABH   ;OPENS A FILE
RWSEQI   EQU     30C6H   ;REW SEQ FILE
INSEQO   EQU     30E7H   ;INIT SEQ OUT FILE
CLSEQO   EQU     3136H   ;CLOSE NEW SEQ FILE
GTBYT    EQU     322CH   ;GET SINGLE BYTE FROM FILE

PTBYT    EQU     324AH   ;PUT BYTE ON FILE
GETTO    EQU     2C0CH   ;GETS A 'TO'

INPCRT   EQU     81C5H   ;JUMP VECTOR NUMBER 31
KBDFL    EQU     81DFH   ;HOLDS NUMBER OF JUMP VECTOR
KBRDY    EQU     81FFH   ;KEYBOARD FLAG (READY)
KBCHA    EQU     81FEH   ;CHAR FROM KEYBOARD
KEYTST   EQU     0024H   ;KEYBOARD SCANNER (KEYTEST)
ADHLA    EQU     3518H   ;ADDS A TO HL WITH CARRYS
CMPHD    EQU     344DH   ;COMPARE FOR HL <=> DE  (C,Z,0)
CMPDH    EQU     3453H   ;COMPARE FRO DE <=> HL  (C,Z,0)
ACRTSP   EQU     0036H   ;TO[illegible]ACK (CPUSP)
LOFL     EQU     [illegible]H   ;[illegible]AG
ESCDX    EQU     32[illegible]   [illegible]
BEGIN    EQU     3768H   ;[illegible]
;
         ELSE
;
PRORG    EQU     0A120H  ;** INTECOLOR 8001  V2.79 & V5.79 **

LO       EQU     1E27H   ;SEND CHAR TO SCREEN
OSTR     EQU     1E58H   ;SEND STRING ENDING WITH 239

EMESS    EQU     2123H   ;FCS ERROR SUBROUTINE
PFSPC    EQU     2D4AH   ;PARSE FILE SPEC
RESET    EQU     219BH
OPEN     EQU     2A6DH   ;OPENS A FILE
RWSEQI   EQU     2D99H   ;REW SEQ FILE
INSEQO   EQU     2DBAH   ;INIT SEQ OUT FILE
CLSEQO   EQU     2E09H   ;CLOSE NEW SEQ FILE
GTBYT    EQU     2EFFH   ;GET SINGLE BYTE FROM FILE
PTBYT    EQU     2F1DH   ;PUT BYTE ON FILE
GETTO    EQU     28B5H   ;GETS A 'TO'

INPCRT   EQU     9F99H   ;JUMP VECTOR NUMBER 31 (INCRT2)
KBDFL    EQU     9FF9H   ;HOLDS NUMBER OF JUMP VECTOR
```

S. P.
ELECTRONIC SYSTEMS
6250 VAN NUYS BLVD.—788-8858
VAN NUYS, CALIFORNIA 91401

```
KBRDY   EQU     9FFCH   ;KEYBOARD FLAG
KBCHA   EQU     9FFBH   ;CHAR FROM KEYBOARD
KEYTST  EQU     21E2H   ;KEYBOARD SCANNER
ADHLA   EQU     1F6FH   ;ADDS A TO HL WITH CARRYS
CMPHD   EQU     1EA4H   ;COMPARE FOR HL <=> DE  (C,Z,0)
CMPDH   EQU     1EAAH   ;COMPARE FOR DE <=> HL  (C,Z,0)
ACRTSP  EQU     000EH   ;TOP OF CRT STACK
LOFL    EQU     9FB4H   ;SYSTEM OUTPUT FLAG
ESCDX   EQU     2000H   ;ESCAPE D ENTRY POINT
BEGIN   EQU     1000H   ;START OF CRT CODE
;
        ENDIF
;
;
;       ** COMMON EQUATES **
;
TIMX3   EQU     0018H   ;KEYBOARD INTERRUPT VECTOR
;
FOPT    EQU     0
FATR    EQU     1
FNAM    EQU     2
FTYP    EQU     8
FVER    EQU     11
FSBK    EQU     12
FSIZ    EQU     14
FLBC    EQU     16
FLAD    EQU     17
FSAD    EQU     19
FSPR    EQU     21
FDBK    EQU     22
FDEN    EQU     23
FAUX    EQU     24
FHAN    EQU     26
FFCN    EQU     28
FDRV    EQU     29
FBLK    EQU     30
FBUF    EQU     32
FXBC    EQU     34
FPTR    EQU     36
EMFN    EQU     15H     ;MISSING FILE NAME
;
;       ** SPECIAL CONTROL CHARACTERS **
;
CTRLB   EQU     02      ;CONTROL B
CTRLC   EQU     03      ;CONTROL C
CR      EQU     13
LF      EQU     10
FF      EQU     12
;
;
; ***********************************
;
        ORG     PRORG
;
;       ** START OF PROGRAM **
;
START:  LXI     SP,STACK        ; SETUP RUNOFF STACK
```

```
;
; *** SETUP FOR 'CI' ***
        MVI     A,31            ; SETUP JUMP VECTOR #31
        STA     KBDFL           ; TO INPCRT
        MVI     A,0C3H          ; LITERALLY 'JMP'
        STA     INPCRT
        LXI     H,CHRINT        ; 'JMP' ADDRESS
        SHLD    INPCRT+1
        XRA     A
        STA     KBRDY           ; READY FOR CHARACTER
; *****
;
        LXI     H,CRLF          ; CR/LF FIRST
        CALL    OSTR
        LXI     H,MSG00         ; PRINT STARTUP MESSAGE
        CALL    OSTR
;
        CALL    SETUP           ; INITIALIZE FILES
;
NEXT:   CALL    BRKTST  ; CHECK FOR 'BREAK' JKP 6/25/79
        JZ      CONINT  ; IF 'Z' FROM 'BRKTST', ABORT
        ;
        CALL    GET
        STA     LINE
        CALL    GET
        STA     LINE+1
        CPI     0
        JZ      LAST    ;HIGH ORDER NEXT ADD 0 ON LAST LINE
        CALL    GET
        STA     LINE+2
        STA     NEW+2
        CALL    GET
        STA     LINE+3
        STA     NEW+3
        LXI     B,LINE+4;BEGINNING OF BASIC LINE
READLP: CALL    GET
        STAX    B       ;STORE CHAR
        CPI     0       ;SEE IF LAST CHARACTER
        JZ      CLEAN   ;IT IS
        INX     B
        JMP     READLP
CLEAN:  LHLD    DROPS   ;RUNNING COUNTER OF REMOVED
        XCHG            ;'DROPS' COUNTER IN DE
        LXI     H,LINE+4;BEGINNING OF OLD LINE
        LXI     B,NEW+4 ;BEGINNING OF NEW LINE
CLOOP:  MOV     A,M
        CPI     34      ;QUOTES
        CZ      QM      ;SPACES WILL STAY
        CPI     ' '
        JZ      SM      ;SPACES WILL GO
        CPI     142     ;REMARK
        JZ      REM
        STAX    B       ;CHARACTER IS GOOD
        CPI     0       ;LAST ONE?
        JZ      FLUSH   ;WRITE LINE
        INX     H
        INX     B
        JMP     CLOOP
SM:     INX     H       ;GO TO NEXT CHAR
        DCX     D       ;UPDATE COUNTER
        JMP     CLOOP
QM:     STAX    B       ;STORE CHARACTER
        INX     H
        INX     B
        MOV     A,M     ;GET NEXT ONE
        CPI     34      ;CLOSING QUOTE?
        RZ              ;YES,GO HOME
        CPI     0       ;END OF LINE?
        RZ              ;GO BACK FOR FLUSH
        JMP     QM
FLUSH:  LHLD    LINE    ;GET OLD ADDRESS
        DAD     D       ;ADJUST FOR REMOVES
        SHLD    NEW
        MOV     H,D     ;HL:=DE
        MOV     L,E
        SHLD    DROPS   ;SAVE ADJUSTMENT
        LXI     B,NEW   ;POINT B AT START
        LDAX    B
        CALL    PUT
        INX     B
        LDAX    B
        CALL    PUT     ;FIRST 4 BYTES
        INX     B
        LDAX    B
        CALL    PUT
        INX     B
        LDAX    B
        CALL    PUT
        INX     B
WRLOOP: LDAX    B
        CALL    PUT
        CPI     0       ;LAST ONE
        JZ      NEXT    ;DO NEXT LINE
        INX     B
        JMP     WRLOOP
LAST:   MVI     A,0
        CALL    PUT
        MVI     A,0
        CALL    PUT
        JMP     EOF
REM:    STAX    B       ;SAVE REM
        INX     B
        INX     H
        MOV     A,M
        CPI     0       ;SEE IF LAST
        JNZ     REM
        STAX    B       ;SAVE LAST
        JMP     FLUSH   ;END OF LINE EXIT
GET:    LXI     H,FPB1
        CALL    GTBYT
        RET
PUT:    LXI     H,FPB2
        CALL    PTBYT
        RET
```

S. P.
ELECTRONIC SYSTEMS
5250 VAN NUYS BLVD.—788-88[illegible]
VAN NUYS, CALIFORNI 914[illegible]

```
;
; ***********************************************************
        ;FCS BREAK TEST    JKP III 6/25/79
        ;
BRKTST: CALL    KEYTST          ; GO SCAN KEYBOARD
        LDA     KBRDY           ; LOOK FOR 'BREAK'
        CPI     50H             ; 'BREAK' KEY
        JZ      BRKTST          ; HANG ON 'BREAK'
        ;
        LDA     KBCHA           ; GET CHAR
        CPI     CTRLB           ; CONTROL B ?
        PUSH    PSW             ; SAVE CHAR
        XRA     A
        STA     KBRDY           ; CLEAR KEYBOARD READY FLAG
        STA     CHARIN          ; CLEAR CHARACTER
        POP     PSW
        JZ      BRKTST          ; HANG ON CONTROL B
        ;
        CPI     CTRLC           ; CONTROL C ?
        RZ                      ; RETURN WITH Z TO ABORT
        CPI     LF              ; LINEFEED ?
        RET                     ; EXIT SETS 'Z' FLAG TO ABORT
        ;
; *******************************************************************
        ;
LOOPE2: JMP     ERROR
;
LOOPE1: JZ      EOF             ;AT END OF FILE - CLOSE OUTPUT
        JMP     ERROR
;
SETUP:  LXI     H,MSG01         ; DISPLAY PROMPT
        CALL    OSTR
        CALL    RESET           ; RESET DISK IF ERROR
        ;
        ;BEGIN USER INPUT SECTION
        ;
        LXI     H,BUFFER        ; POINT AT BUFFER
L02:    CALL    CI              ; READ FROM CONSOLE
        CPI     CTRLC           ; IS IT A CTRL C ?
        JZ      CONINT          ;* YES, EXIT FOR CONSL INTERRUPT
        CPI     13              ; IS IT CR ?
        JZ      X02             ; YES, GO PROCESS CR
        CPI     26              ; IS IT BS ?
        JZ      G021            ; YES, GO PROCESS BACKSPACE
        ;
        CPI     11              ; ERASE LINE ?
        JZ      SETUP           ; PROMPT AGAIN
        CPI     127             ; DELETE CHAR ?
        JZ      G021            ; SAME AS BACKSPACE
        ;
        CPI     91              ; IS IT A 'Z' OR LESS ?
        JNC     L02             ; IF NOT, BAD CHAR
        CPI     32              ; IS IT 'SPACE' OR GREATER ?
        JC      L02             ; IF NOT, BAD CHAR
        ;
        MOV     M,A             ; STORE CHARACTER
        CALL    LO              ; ECHO CHARACTER
```

```
         INX     H               ; BUMP POINTER
         JMP     L02             ; CONTINUE LOOP
         ;
G021:    LXI     D,BUFFER        ; GET ADDR OF 'BUFFER'
         CALL    CMPDH           ; IS POINTER AFTER BUF BEGIN
         JNC     SETUP           ; IF NOT, PROMPT AGAIN
         MVI     A,26            ; 'BACKSPACE'
         CALL    LO
         MVI     A,' '           ; SPACE OVER LAST CHAR
         CALL    LO
         MVI     A,26            ; ANOTHER 'BACKSPACE'
         CALL    LO
         DCX     H
         JMP     L02             ; CONTINUE LOOP
         ;
X02:     MVI     M,0             ; INSERT TERMINATOR
         MVI     A,CR            ; PRINT CARRIAGE RETURN
         CALL    LO
         MVI     A,LF            ; LINEFEED SEQUENCE
         CALL    LO
         ;
         ;END OF INPUT SECTION
         ;
         LXI     H,BUFFER        ; POINT AT BUFFER
         LXI     D,FPB1          ; POINT AT INPUT FPB
         LXI     B,DEFAULT       ; POINT AT DEFAULT TYPE
         CALL    PFSPC           ; PARSE FILE SPEC
         JC      E02             ; IF CARRY THEN ERROR
         ;
         PUSH    H               ;SAVE POINTER TO EOL OR 'TO'
         ;                                   JKP 6/28/79
         LXI     H,FPB1          ; POINT AT INPUT FPB
         MVI     A,0             ; SETUP AS 'OLD' FILE
         MOV     M,A
         CALL    OPEN            ; OPEN THE 'OLD' FILE
         JC      E02
         ;
         LXI     H,FPB1          ; COPY FPB1 TO FPB2
         LXI     D,FPB2
         MVI     B,38    ;MAX NUMBER OF CHAR IN FILE SPECS
L03:     MOV     A,M
         STAX    D
         INX     H
         INX     D
         DCR     B
         JNZ     L03
         ;
         LXI     H,FPB2+FTYP     ; POINT AT OUTPUT TYPE
         MVI     M,'B'           ; STORE 'B'
         INX     H
         MVI     M,'A'           ; STORE 'A'
         INX     H
         MVI     M,'S'           ; STORE 'S'
         INX     H
         ;
         LXI     D,FPB1+FVER     ;POINT AT OLD VERSION NUMBER
         LDAX    D
```

```
         INR     A               ; NEXT 'VERSION' NUMBER
         MOV     M,A             ;STORE NEW VERSION NUMBER
         ;
         POP     H               ;RESTORE POINTER TO EOL OR 'TO'
         MOV     A,M             ;SEE IF EOL     JKP 6/28/79
         ANA     A
         CNZ     NWNAME          ;PARSE NEW NAME
         ;
         LXI     H,FPB2          ; POINT AT OUTPUT FILE
         MVI     A,1             ; SETUP AS 'NEW' FILE
         MOV     M,A
         CALL    OPEN            ; OPEN THE 'NEW' FILE
         JC      E02
         LXI     H,IBUF          ; POINT AT INPUT BUFFER
         SHLD    FPB1+FBUF       ; SAVE INPUT BUFFER ADDRESS
         LXI     H,OBUF          ; POINT AT OUTPUT BUFFER
         SHLD    FPB2+FBUF       ; SAVE OUTPUT BUFFER ADDRESS
         LXI     H,1024          ; SETUP BUFFER SIZE
         SHLD    FPB1+FXBC       ; SAVE INPUT BUFFER SIZE
         SHLD    FPB2+FXBC       ; SAVE OUTPUT BUFFER SIZE
         LXI     H,FPB1          ; POINT AT INPUT FPB
         CALL    RWSEQI          ; REWIND INPUT FILE
         LXI     H,FPB2          ; POINT AT OUTPUT FPB
         CALL    INSEQO          ; INITIALIZE OUTPUT FILE
         RET                     ; RETURN
         ;
         ;ADD FOR DIFFERENT SOURCE AND DESTINATION FILE NAMES
         ;  JKP III  6/28/79
NWNAME:  CALL    GETTO           ; LOOK FOR 'TO'
         JNZ     NWNMER          ; IF NO 'TO', ERROR
         LXI     D,FPB2          ; POINT AT NEW FPB
         LXI     B,DEFAULT       ; POINT AT DEFAULT OUTPUT TYPE
         CALL    PFSPC           ; PARSE NEW NAME
         RNC                     ; NO ERROR
         MOV     A,B             ; GET ERROR CODE
         CPI     EMFN            ; 'MISSING FILE NAME'
         RZ                      ; USE OLD NAME, DEFAULT TYPE
NWNMER:  POP     H               ; POP OFF RETURN ADDRESS
         JMP     E02             ; PRINT ERROR AND RETRY
         ;
; ******************************************************
         ;CHARACTER INTERRUPT ROUTINE
         ;THIS IS VECTORED TO FROM THE KEYBOARD
         ;INPUT ROUTINE THROUGH 'INPCRT'   JKP  6/25/79
         ;
CHRINT:  PUSH    H
         PUSH    PSW
         LXI     H,CHARIN        ; LAST CHAR FROM KEYBOARD
         XRA     A
         CMP     M               ; IS LAST CHAR NON-ZERO ?
         JNZ     CFIN            ; LAST CHAR NOT YET TAKEN
         MOV     A,E             ; CHARACTER
         ANI     127             ; MAKE ASCII
         MOV     M,A             ; STORE ASCII CHAR
         ;
CFIN:    POP     PSW
         POP     H
```

```
        EI                      ; ENABLE INTERRUPTS
        RET
        ;
CHARIN: DB      0               ; INPUT CHAR STORAGE (INIT 0)
        ;
; ****************************************
        ;CI ROUTINE
        ;THIS READS A CHARACTER
        ;AND ESCAPES FROM A HUNG PROGRAM
        ;
CI:     EI              ;ENABLE INTERRUPTS
        LDA     CHARIN  ;GET CHAR
        CPI     0       ;HAVE CHARACTER?
        JZ      CI      ;IF NOT, HANG FOR CHARACTER
        ;
        PUSH    PSW     ;SAVE CHAR
        XRA     A
        STA     KBRDY   ;CLEAR KEYBOARD READY FLAG
        STA     CHARIN  ;CLEAR CHAR 'BUFFER'
        POP     PSW
        RET             ;ECHO CHAR FROM MAIN ROUTINE
        ;
; ****************************************
;
E02:    CALL    EMESS           ; EMIT ERROR MESSAGE
        JMP     SETUP           ; READ NEW LINE
;
ERROR:  LXI     H,MSG02         ; PRINT ERROR MESSAGE
        CALL    OSTR
        MVI     B,0             ; SETUP NO ERROR
        ;
EXIT:   XRA     A
        STA     LOFL    ; TURN OFF PRINTER
        LHLD    ACRTSP  ; GET STACK POINTER
        SPHL
        LXI     H,KBDFL ; POINT HL AT INPUT FLAG
        LXI     D,BEGIN ; START OF CRT CODE
        PUSH    D
                        ;* ASSUME POWER-UP FLAG IS SET
        JMP     ESCDX   ; JUMP TO ESC D CODE (FCS)
;
EOF:    LXI     H,FPB2          ; POINT AT OUTPUT FILE
        CALL    CLSEQO          ; CLOSE OUTPUT FILE
        MVI     B,0             ; SETUP NO ERROR
        JMP     EXIT
;
CONINT: LXI     H,MSG03         ; CONSOLE INTERRUPT MSG
        CALL    OSTR
        MVI     B,0             ; SETUP NO ERROR
        JMP     EXIT            ; GO EXIT
;
;
;       ***  MESSAGES  ***
;
        IF CRTYPE EQ 2
MSG00:  DB      18,11,'COMPUCOLOR II V6.78 '
        ELSE
```

```
MSG00:  DB      18,11,'INTECOLOR 8001 V2.79 & V5.79 '
        ENDIF
;
        DB      21,'SPACE REMOVER ',22,'V1.00'
;
         IF EXPMNT EQ 1
         DB     17,' EXPERIMENTAL'
         ENDIF
;
        DB      13,10,239
;
;
MSG01:  DB      19,11,'COMPACT>',18,239
MSG02:  DB      17,'HARDWARE ERROR !',13,10,239
MSG03:  DB      13,10,17,'CONSOLE INTERRUPT !',13,10,239
CRLF:   DB      13,10,239
;
DEFAULT:DB      'BAS'
;
EOC     EQU     $
;
        .IFP1
        .PRINT  'SAVE FROM ',PRORG,' TO',$
        .PRINT  ''
        ENDIF
;
;       *** DATA AREAS ***
;
BUFFER: DS      256
IBUF:   DS      1024    ;INPUT BUFFER
OBUF:   DS      1024    ;OUTPUT BUFFER
;
        DS      100     ;STACK AREA
STACK:
        DS      2
;
FCSSP:  DS      2       ;FCS STACK POINTER
FPB1:   DS      38      ;INPUT FPB
FPB2:   DS      38      ;OUTPUT FPB
LINE:   DS      100     ;LINE STORE
DROPS:  DB      0,0     ;'DROPS' COUNTER
NEW:    DS      100     ;NEW LINE
;
EOP     EQU     $
;
        END     START
```

STARTREK SP71

A super version of Startrek. This version includes such items as DOC, NAV, SRS, LRS, PHA, TOR, SHE, DAM, COM. The computer contains 6 different commands and the program is well illistrated. In addition to the super TREK program OIL a board type game also appears. You may note from the menu that the disk is very full with only 13 block left over

PRICE $20.00

```
DIRECTORY CD0: STARTREK 06        SP71

ATR  NAME TYPE VR SBLK SIZE LBC LADR SADR

 03 SSKDSP.DSP;01 0006 0020 80  7000 7000
 03 SSKINS.BAS;01 0026 0047 7E  C29A A618
 03 SUPSK2.BAS;01 006D 008F 34  829A C9CE
 03 OIL   .BAS;01 00FC 0059 1D  829A AEB7
 03 OILCO .DSP;01 0155 0020 80  7000 7000
 03 MENU  .BAS;01 0175 0008 4E  829A 8668
 01 <FREE SPACE>  017D 0013
```

INVENTORY & INVOICEING SP29

This is another one we can't guarentee but we have traded and sold (at a much higher price) a number of copies of it already. In the past, in general, we have not talked this program up much but have saved it for the people who purchased their machines thru us. It is a very extensive program with quite a number of files and at the price being offered it is quite a bargain.

PRICE $40.00

```
DIRECTORY CD0: INVENTORY  03

ATR  NAME TYPE VR SBLK SIZE LBC LADR SADR

 03 MENU  .BAS;01 0003 006F 0E  829A B9A8
 03 UIN   .RND;01 0072 0032 80  0064 0040
 03 INV   .RND;01 00A4 0032 80  0064 0040
 03 NBB   .RND;01 00D6 001E 80  0064 0024
 03 BOF   .RND;01 00F4 001E 80  0064 0024
 03 NIN   .RND;01 0112 000E 80  0019 0040
 03 BOR   .RND;01 0120 001E 80  0064 0024
 01 <FREE SPACE>  013E 0052
```

I am generally not in favor of presenting program printouts in this bulletin unless they either represent unusual ISC related programing or demonstrate the use of internal machine locations. What we are going to print in the next few issues are source codes for some of the most popular ISC programs. We have a large number of these in our files and in many cases we even have the origonal source codes upon which ISC based many of their programs. (This is because many of these programs were part of out library before ISC started selling them for their machines) Where possible we will present the ISC version as our intent is to demonstrate how to use ISC machines and show where various subroutines are located.

INVENTORY #SP 29

```
130 PRINT TAB( 10);"1-LOOK UP THE FILES.":PRINT
140 PRINT TAB( 10);"2-ADD OR DELETE ITEMS ON INVENTORY FILE.":PR
INT
150 PRINT TAB( 10);"3-MODIFY THE FILES.":PRINT
160 PRINT TAB( 10);"4-RECEIVE MERCHANDIZE.":PRINT
170 PRINT TAB( 10);"5-DO INVOICING.":PRINT
180 PRINT TAB( 10);"6-ORDER MERCHANDIZE.":PRINT
190 PRINT TAB( 10);"7-TERMINATE THE PROGRAM.":PRINT
200 INPUT "YOUR SELECTION PLEASE: ";S
205 I= 0:J= 0
210 IF S> 7OR S< 1THEN GOTO 200
220 S= INT (S):ON SGOTO 250,1810,2560,2950,3480,6310,7000
250 PLOT 6,2,12,15
260 PRINT "YOU CAN LOOK-UP THE FOLLOWING FILES: "
270 PRINT :PRINT TAB( 10);"1-INVENTORY FILE."
280 PRINT :PRINT TAB( 10);"2-BACK-ORDER FILE."
290 PRINT :PRINT TAB( 10);"3-BACK-ORDER RELEASE FILE."
295 PRINT :PRINT TAB( 10);"4-RETURN TO MAIN PROGRAM."
300 PRINT :INPUT "           YOUR SELECTION PLEASE: ";S
310 IF S> 4OR S< 1THEN GOTO 300
312 S= INT (S):ON SGOTO 315,1150,1630,100
315 PLOT 6,2,12,15
330 PRINT "INVENTORY FILE."
340 PRINT :PRINT "YOU CAN 1-LIST THE COMPLETE FILE."
350 PRINT :PRINT TAB( 9);"2-LIST THE FILE FROM ? TO ?."
360 PRINT :PRINT TAB( 9);"3-LIST A PARTICULAR ITEM."
370 PRINT :PRINT TAB( 9);"4-COMPUTE INVENTORY VALUE AT COST."
380 PRINT :PRINT TAB( 9);"5-COMPUTE INVENTORY AT RETAIL."
390 PRINT :PRINT TAB( 9);"6-RETURN TO MAIN PROGRAM."
1160 PRINT "BACK ORDER FILE"
1170 PRINT :PRINT "YOU CAN 1-LIST THE COMPLETE B/O FILE."
1180 PRINT :PRINT TAB( 9);"2-HOW MANY OF AN ITEM IS ON BACK ORDE
R AND TO WHOM."
1640 PRINT "BACK ORDER RELEASE FILE.":PRINT
1650 PRINT "THIS FILE CONTAINS CUSTOMERS FOR WITCH ITEMS HAVE BE
EN "
1660 PRINT "RECEIVED BUT INVOICES HAVE NOT BEEN PRINTED YET."

1905 IF PN$= "0"THEN GOTO 1810
1910 PRINT :PRINT
1920 INPUT "DESCRIPTION:              ";DE$
1930 INPUT "UNIT OF MEASURE:          ";UM$
1940 INPUT "QUANTITY ON HAND:         ";QH
1950 INPUT "QUANTITY ON ORDER:        ";QO
1960 INPUT "COST:                     ";CS
1970 INPUT "RETAIL:                   ";RT
1980 INPUT "MINIMUM QUANTITY:         ";MI
1990 INPUT "MAXIMUN QUANTITY:         ";MQ
2000 INPUT "REORDER QUANTITY:         ";RQ
2010 INPUT "YEAR TO DATE SALE:        ";YS
2020 PRINT :INPUT "IS ALL DATA CORRECT? (Y OR N)";A$
```

S. P.
ELECTRONIC SYSTEMS
6250 VAN NUYS BLVD.—788-8858
VAN NUYS, CA 91401

LEDGER appears to be an excellent ledger program but we have not run it so we can not gaurentee it. It contains about 600 lines of code and represents a prodigious effort. From those to whom we have already given a copy of this program we have reason to expect that it either works as is or with little modification. It certainly could be boiled down a bit with better coding but at the price it represents one hell of a bargain.
The program contains the following sections in addition to the coding listed below. PAYROL, EXPENSES, ACCOUNTS PAYABLE, CASH ENTRIES, PROFIT AND LOSS, AND PETTY CASH

PRICE $35.00

SP 90

```
DIRECTORY CD0: LEGER1       03

ATR  NAME TYPE VR SBLK SIZE LBC LADR SADR

 03 MENU  .BAS;01 0003 007A 14  829A BF2E
 01 <FREE SPACE>  007D 0113
```

```
200 DATA "SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY","FRI
DAY","SATURDAY"
220 DATA 100,"MERCHANDISE",200,"SHIPPING",300,"FREIGHT"
230 DATA 400,"POSTAGE",1000,"BAD DEBTS",1500,"SHOP EXPENSE"
240 DATA 2000,"ACCOUNTING",2100,"INSURANCE",2200,"LEGAL"
250 DATA 2300,"LICENSES",2400,"OFFICE EXPEN",2500,"MISC/PETTY"
260 DATA 2600,"LEASING EXPS",2700,"LEASING IN",2800,"TELEPHONE"
270 DATA 2900,"UTILITIES",3000,"PAYROLL",3100,"TAX-SOC.SEC."
280 DATA 3200,"TAX-ST UNEM.",3300,"INTEREST",3400,"TAX-OTHER"
290 DATA 3500,"CONTRIBUTIONS",4000,"ENTERTAINMNT",4100,"ADVERTIS
ING"
300 DATA 4200,"TRAVEL",4300,"AUTO",4400,"DUES+SUBSCRP"
310 DATA 5000,"NOTES+LOANS",5100,"FED.INC.TAX",5200,"LOAN RECVBL
"
320 DATA 6000,"PERSONAL ACS",7000,"FIXED ASSETS"
350 PRINT :PRINT :PRINT
360 INPUT "DEPARTMENT NUMBER .... ";K1
365 PRINT
370 INPUT "F I R M   N A M E .... ";K1$
380 PRINT
390 INPUT "FOR  WEEK  ENDING .... ";WE$
620 PRINT "ENTER THE FOLLOWING ITEMS FOR EACH NEW EMPLOYEE"
630 PRINT "  ITEMS SHOULD BE SEPARATED BY COMMAS"
635 PRINT
640 FOR I= 1TO N2
650 PRINT "EMPL#,  S.S.#,  EMPL. NAME"
660 INPUT E0$,E1$,E$
670 PRINT "HOURLY WAGE,  DEPENDENT CODE,  DEDUCT.AMT,  MISC DED.
AMT"
680 INPUT E0,D,E1,E2
690 PRINT "GROSS YEAR-TO-DATE (YTD),  FICA YTD,  FED TAX YTD,  S
TATE YTD,  MISC.DED YTD"
700 INPUT YO,Y1,Y2,Y3,Y4
710 Y5= Y0- Y1- Y2- Y3- Y4
720 PRINT "GROSS THIS PERIOD (SINCE LAST P+L STATEMENT), TAXES T
HIS PERIOD"
730 INPUT P0,P1
740 PRINT "GROSS THIS QTR,  FICA THIS QTR,  FED.TAX THIS QTR"
2370 PRINT "ENTER LEDGER ITEMS IN ORDER...."
2380 PRINT " ACCT.NO., CHECK NO., DATE, PAYEE, AMOUNT"
2385 PRINT
2390 REM
```

The Stolersoft disk is a double disk containing two disk full of programs submitted to us by a member in Jugoslavia It contains 11+ games of varying quality.

SP 69

PRICE $25.00

```
                M   E   N   U
      =========

 1 ) WUMPUS ADVENTURE

 2 ) GOLD MINE

 3 ) PRESS UPS

 4 ) STAR-WARS

 5 ) BLACK BOX

 6 ) CAMEL ADVENTURE

 7 ) LOGIC CIRCUIT ANALISE

WHICH PROGRAM DO YOU WANT ???
```

```
DIRECTORY CD0: STOLERSOFT 04

ATR  NAME TYPE VR SBLK SIZE LBC LADR SADR

 03 WUMPUS.BAS;01 0004 003A 7C  829A 9F96
 03 GOLD  .BAS;01 003E 001D 6C  829A 9106
 03 PRESS .BAS;01 005B 0023 27  829A 93C1
 03 GREED .BAS;01 007E 0015 79  829A 8D13
 03 BLABOX.BAS;01 0093 0015 1F  829A 8CB9
 03 STRWRS.BAS;01 00A8 0019 22  829A 8EBC
 03 LCA   .BAS;01 00C1 001F 04  829A 919E
 03 MENU  .BAS;01 00E0 0005 35  829A 84CF
 03 CAMEL .BAS;01 00E5 0022 57  829A 9371
 03 INDI  .BAS;01 0107 0010 2F  829A 8A49
 03 LANDER.PRG;01 0117 0002 23  9E00 9E00
 03 LANDER.BAS;01 0119 0013 69  829A 8C03
 03 FLIGHT.BAS;01 012C 0017 2A  829A 8DC4
 03 DANCE .PRG;01 0143 0002 23  FE00 FE00
 03 DANCE .BAS;01 0145 000E 22  829A 893C
 01 <FREE SPACE>  0153 003D
```

SP69

```
   M E N U
  =========

1 ) INDI 500 - A CAR DRIVING GAME

2 ) SPACE FLIGHT

   3 ) SPACECRAFT LANDER

   4 ) DANCE - ANIMATION
```

S. P.
ELECTRONIC SYSTEMS
5250 VAN NUYS BLVD.—788-8858
VAN NUYS, CALIFORNI 91401

SCHAFERINV SP 58
Seems to be a good working inventory program used elsewhere and translated to the CCII. It has been partially tested but is NOT GAURENTEED but at the price, for an inventory program, it again a bargain.

```
DIRECTORY CD0: SCAFERINV  06

ATR  NAME TYPE VR SBLK SIZE LBC LADR SADR

 03 INVNT1.RND;01 0006 0014 80  0014 0080
 03 MENU  .BAS;01 001A 0039 7E  829A 9F18
 03 ~~INV~~   .~~BAS;01~~ 0053 003A 2F  829A 9F49
 01 <FREE SPACE> ~~000D~~ ~~01~~03
```

SP58 ~~35~~

SP 58
$35

WE HAVE NOTED A NUMBER OF OTHER BULLETINS ARE GIVING LESSONS IN ASSEMBLE LANGUAGE. WE HAVE CONSIDERED DOING SO ALSO BUT BECAUSE OF THE GREAT NUMBER OF EXCELLENT BOOKS ALREADY AVAILABLE ON THE SUBJECT WE ARE HOLDING OFF FOR THE PRESENT.

```
100 PLOT 12:REM  NEW KEYBOARD ROUTINE.                    RAY SCHREINER

110 REM FOR EXTENDED DISK BASIC (EDB) (42B) VERSION 2/1/78
120 REM CPU O.S. VERSION  11/29/77
130 REM FCS VERSION (2.0) 11/29/77

140 PRINT "ONCE THIS ROUTINE IS POKED INTO MEMORY, IT CAN BE SAVED ON THE DISK,
150 PRINT "AND LOADED IN BY ANY PROGRAM THAT MIGHT WANT TO USE IT. JUST SET THE
160 PRINT "USER JUMP TO FFCO (195,192,255) AND THEN PLOT27,30. THE KEY STRUCK
170 PRINT "WILL BE AT 9FFEH: PEEK(-24578). IF NO KEY WAS STRUCK, 9FFE WILL
180 PRINT "BE A '7'. ESCAPE & CONTROL CODES will not be EXECUTED.

190 X= - 24902:FOR I= OTO 2:READ Y:POKE X+ I,Y:NEXT :REM (9EBAH)
200 DATA 195,192,255:REM SET USER JUMP VECTOR TO HEX ADDRESS FFCO

210 X= - 64:FOR I= OTO 52:READ Y:POKE X+ I,Y:NEXT :REM POKE ROUTINE

220 DATA 33,227,255:REM     LXI  H, TABLE     ;POINT TO ROW TABLE
230 DATA 126:REM            MOV  A, M         ;1st TABLE ENTRY IS A FLAG
240 DATA 50,223,159:REM     STA  9FDFH        ;SET FLAG INDICATING "NO ECHO"
250 DATA 50,254,159:REM     STA  9FFEH        ;INDICATE TO BASIC "NO KEY STRUCK"
260 DATA 35:REM         X1: INX  H            ;INCREM. PNTR TO NEXT ROW NUMBER
270 DATA 126:REM            MOV  A, M         ;FETCH ROW NUMBER
280 DATA 254,64:REM         CPI  64           ;HAVE WE CHECKED ALL ROWS ?
290 DATA 202,218,255:REM    JZ   EXIT         ;YES, RETURN TO BASIC. NO KEY HIT
300 DATA 211,7:REM          OUT  07           ;NO, SELECT ROW
310 DATA 219,1:REM          IN   01           ;GET ANY PENDING CHARACTER
320 DATA 254,255:REM        CPI  -1           ;ANYTHING THERE ?
330 DATA 202,202,255:REM    JZ   X1           ;IF NOT, POLL NEXT ROW
340 DATA 205,24,0:REM EXIT:CALL 0018H         ;YES, DECODE TO ASCII
350 DATA 62,23:REM          MVI  A, 23        ;SET FLAG INDICATING "BASIC"
360 DATA 50,223,159:REM     STA  9FDFH        ;SET THE FLAG
370 DATA 201:REM            RET               ;TO BASIC

380 DATA 7,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64:REM ROW TABLE

390 REM PROGRAM TO SHOW KEY STRUCK but NOT ECHOED, AND THAT
400 REM YOU CAN DO SOMETHING ELSE WHILE WAITING FOR A HIT.

410 FOR I= 1TO 10000:REM                  LOOK 10,000 TIMES
420 A= RND (1):REM                        PULL OUT A RANDOM NUMBER
430 PLOT 27,30:REM                        CALL THE ROUTINE (POLL THE ENTIRE KB)
440 A= PEEK (- 24578):REM (9FFEH)         SEE IF A KEY WAS STRUCK
450 IF A= 7THEN NEXT I:GOTO 410:REM       NO HIT, PULL 1 MORE. IF DONE, RESTART
460 PRINT A;I:GOTO 410:REM                GOT A HIT. PRINT IT, LOOP # & RESTART
```

S. P.
ELECTRONIC SYSTEMS
5250 VAN NUYS BLVD.—788-8850
VAN NUYS, CALIFORNI 91401